Interactive Computing and Processing of NASA Land Surface Observations using Google Earth Engine

Andrew Molthan[1], Jason Burks[1], and Jordan Bell[2]

[1]Earth Science Office / NASA Marshall Space Flight Center, Huntsville, AL

[2]University of Alabama in Huntsville, Huntsville, Alabama

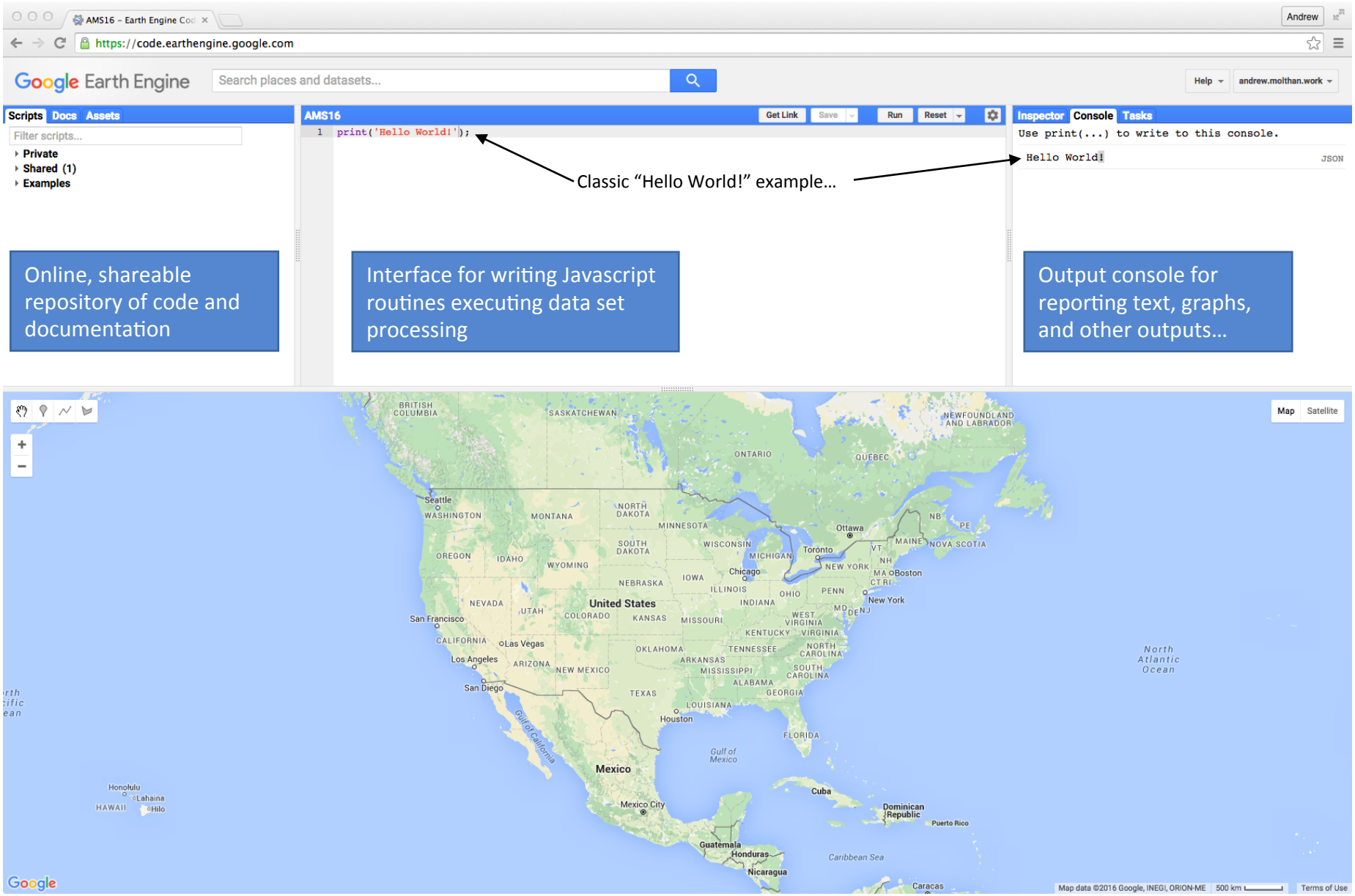
andrew.molthan@nasa.gov

Presentation 8.2, 32nd Conference on Environmental Information Processing Technologies (EIPT) 2016 (96th) AMS Annual Meeting, New Orleans, LA

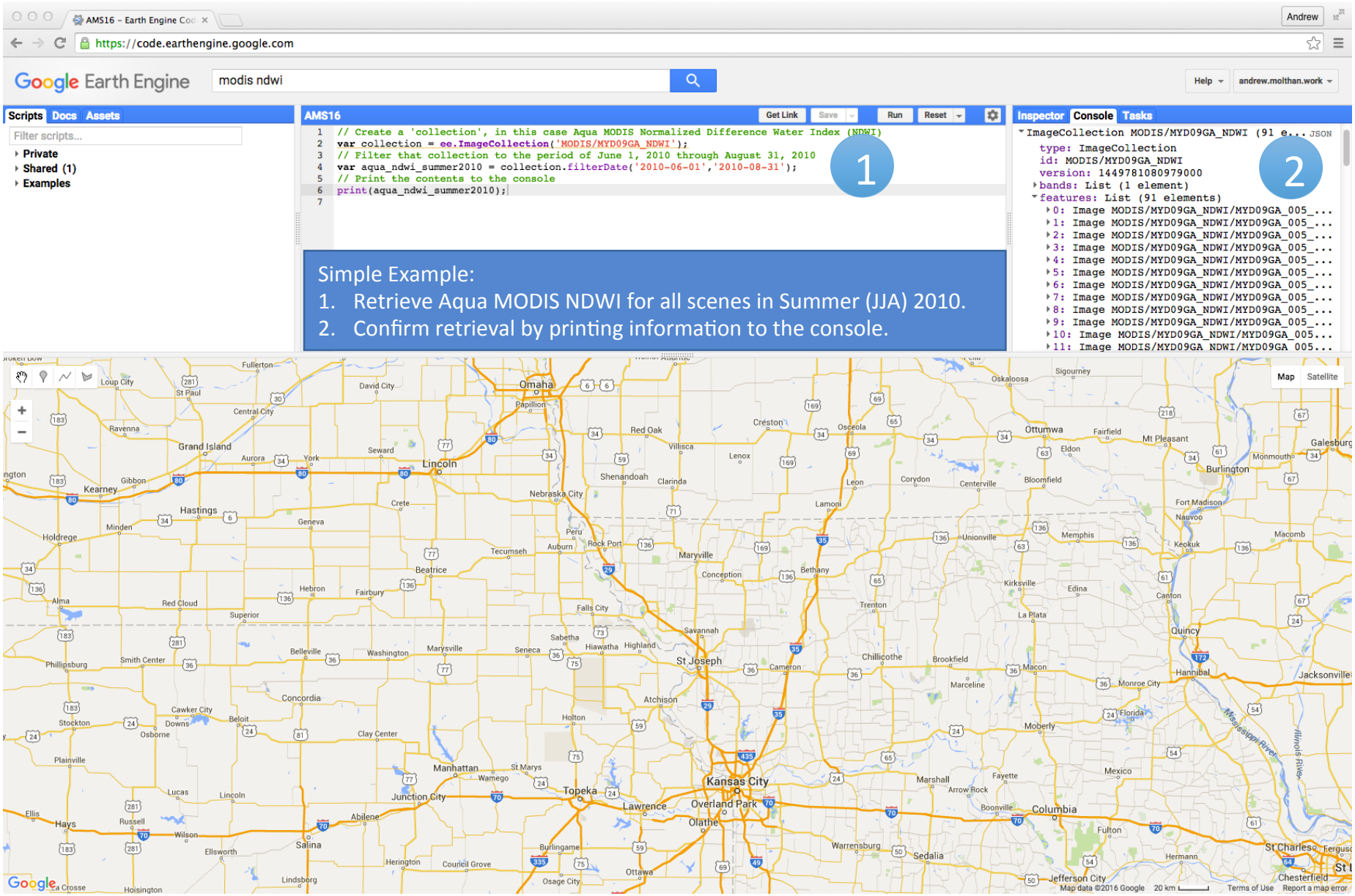
Background

- Google's Earth Engine offers a "big data" approach to processing large volumes of NASA and other remote sensing products. https://earthengine.google.com/
- Interfaces include a Javascript or Python-based API, useful for accessing and processing over large periods of record for Landsat and MODIS observations.
  - Other data sets are frequently added, including weather and climate model data sets, etc.
- Demonstrations here focus on exploratory efforts to perform land surface change detection related to severe weather, and other disaster events.

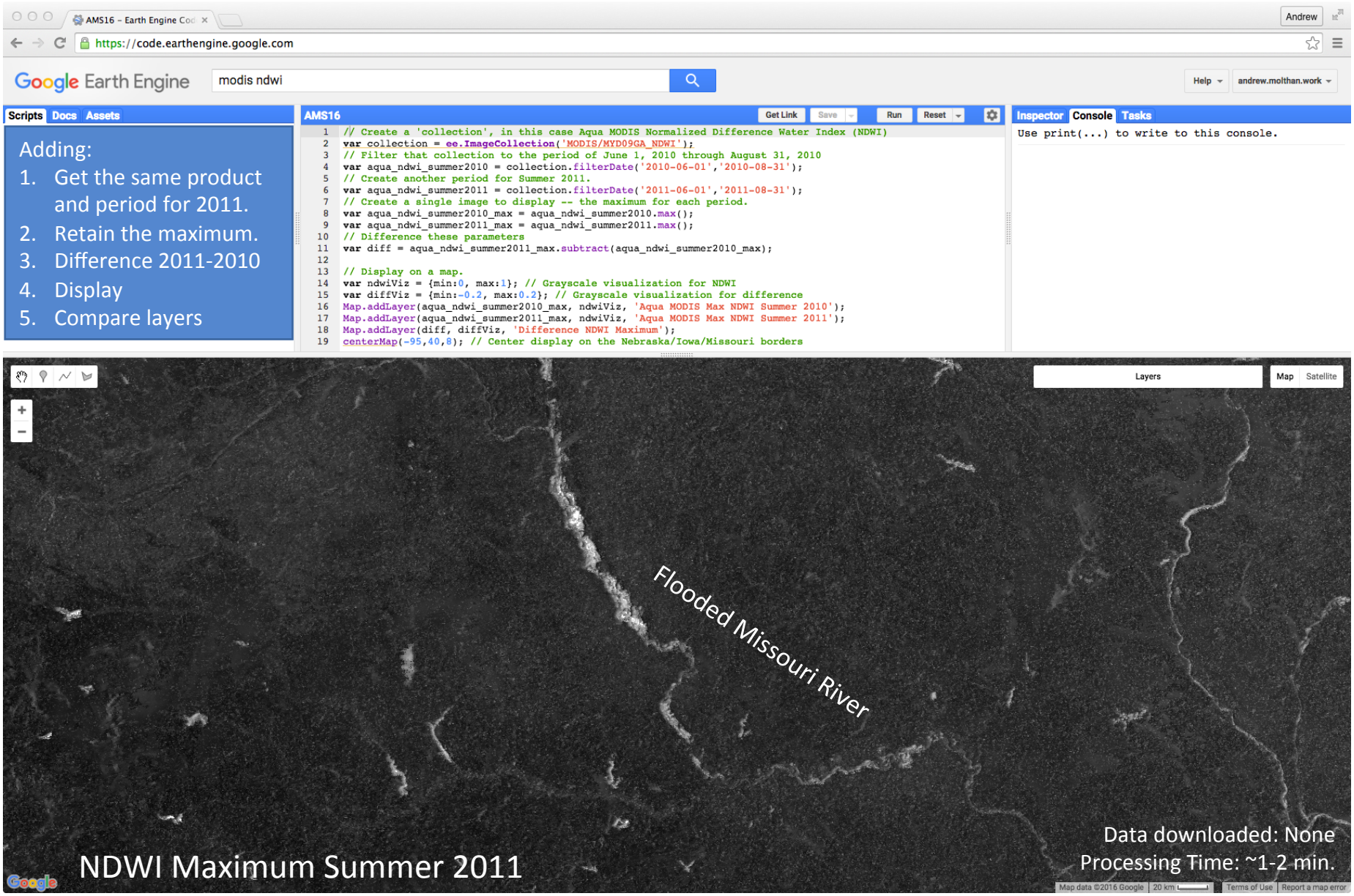
Javascript API (http://ee-api.appspot.com)



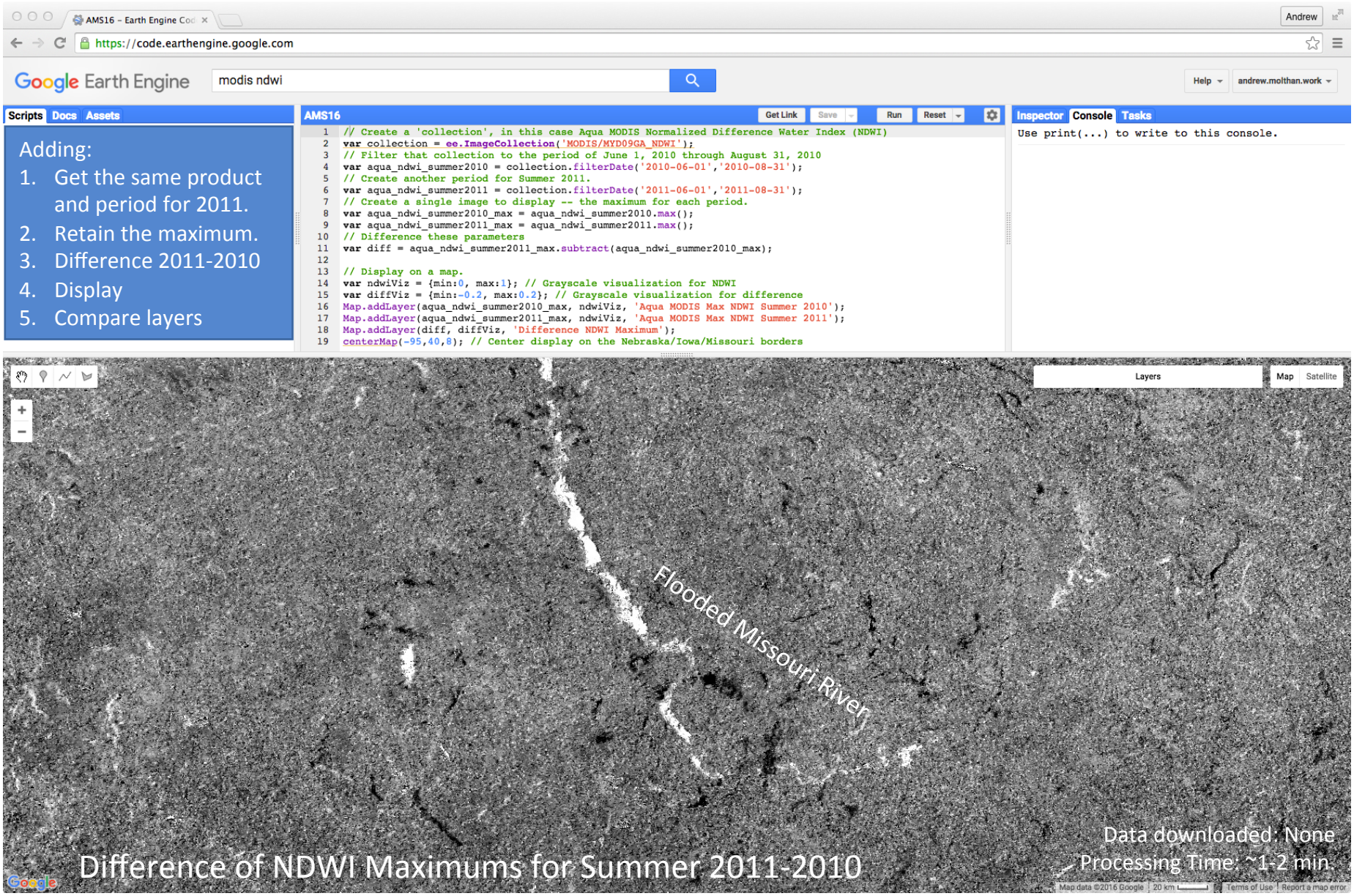
Javascript API (http://ee-api.appspot.com)



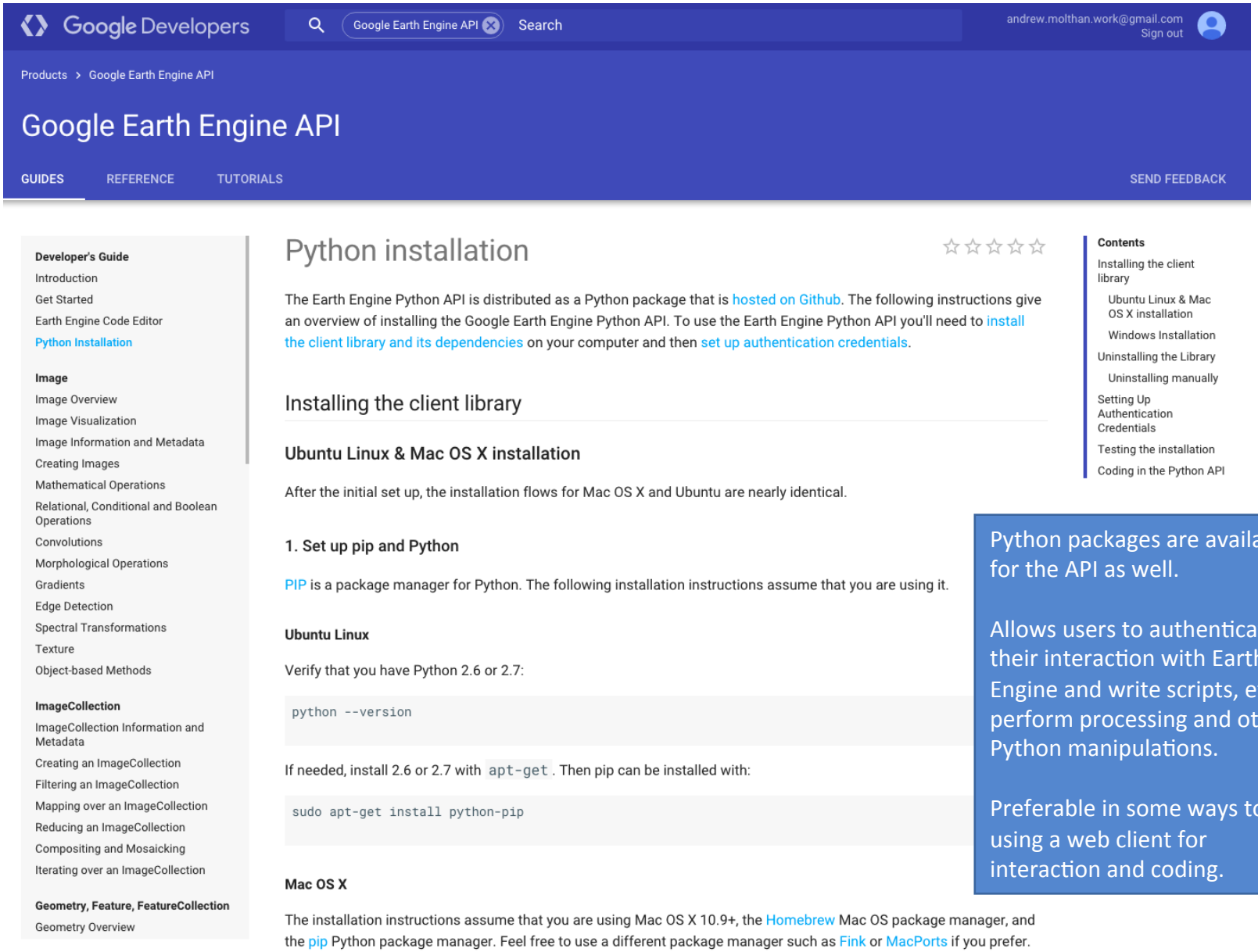
Javascript API (http://ee-api.appspot.com)



Javascript API (http://ee-api.appspot.com)



Python API (https://developers.google.com/earth-engine/python_install?hl=en)

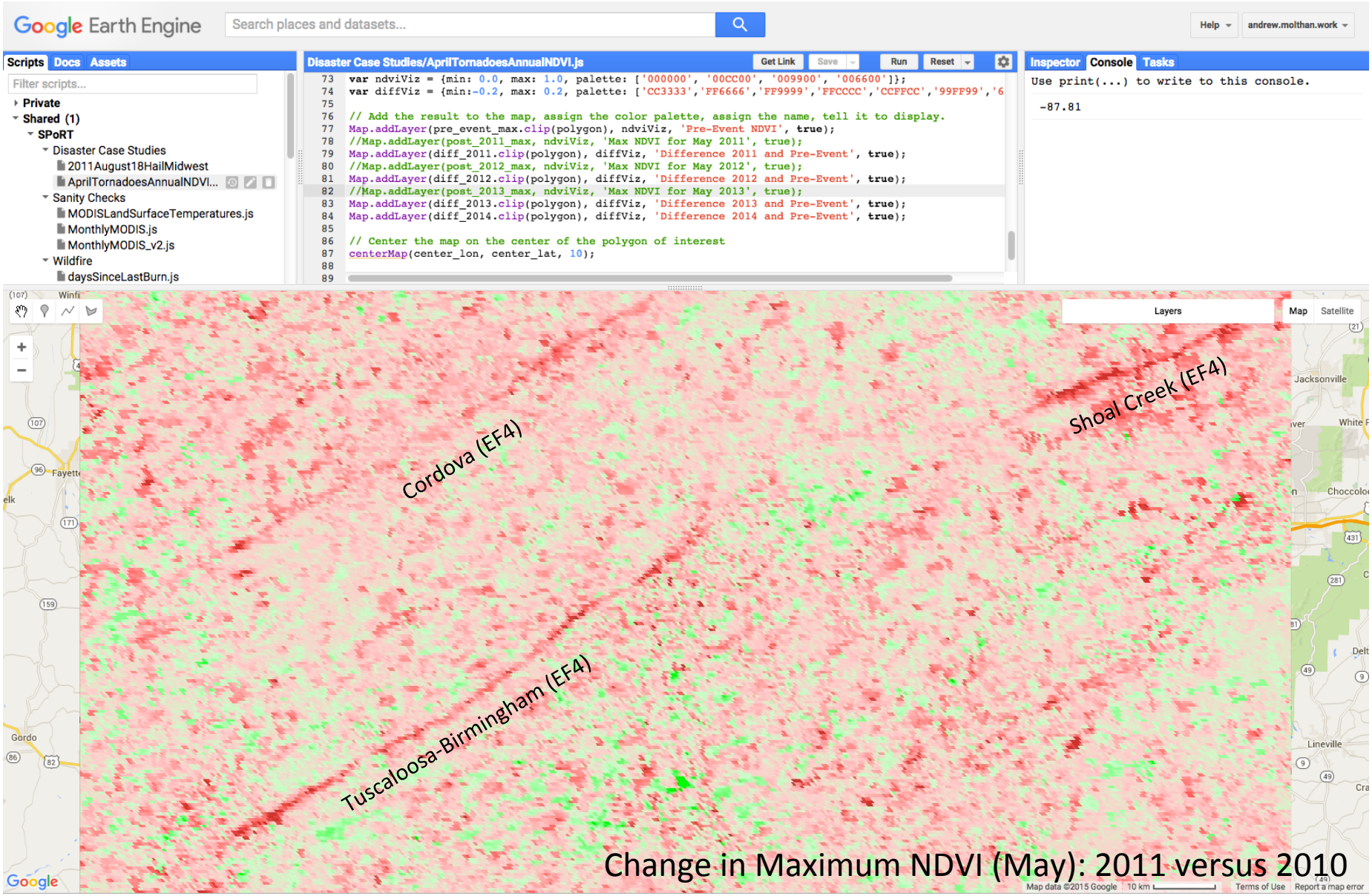


Python packages are available for the API as well.

Allows users to authenticate their interaction with Earth Engine and write scripts, etc. to perform processing and other Python manipulations.

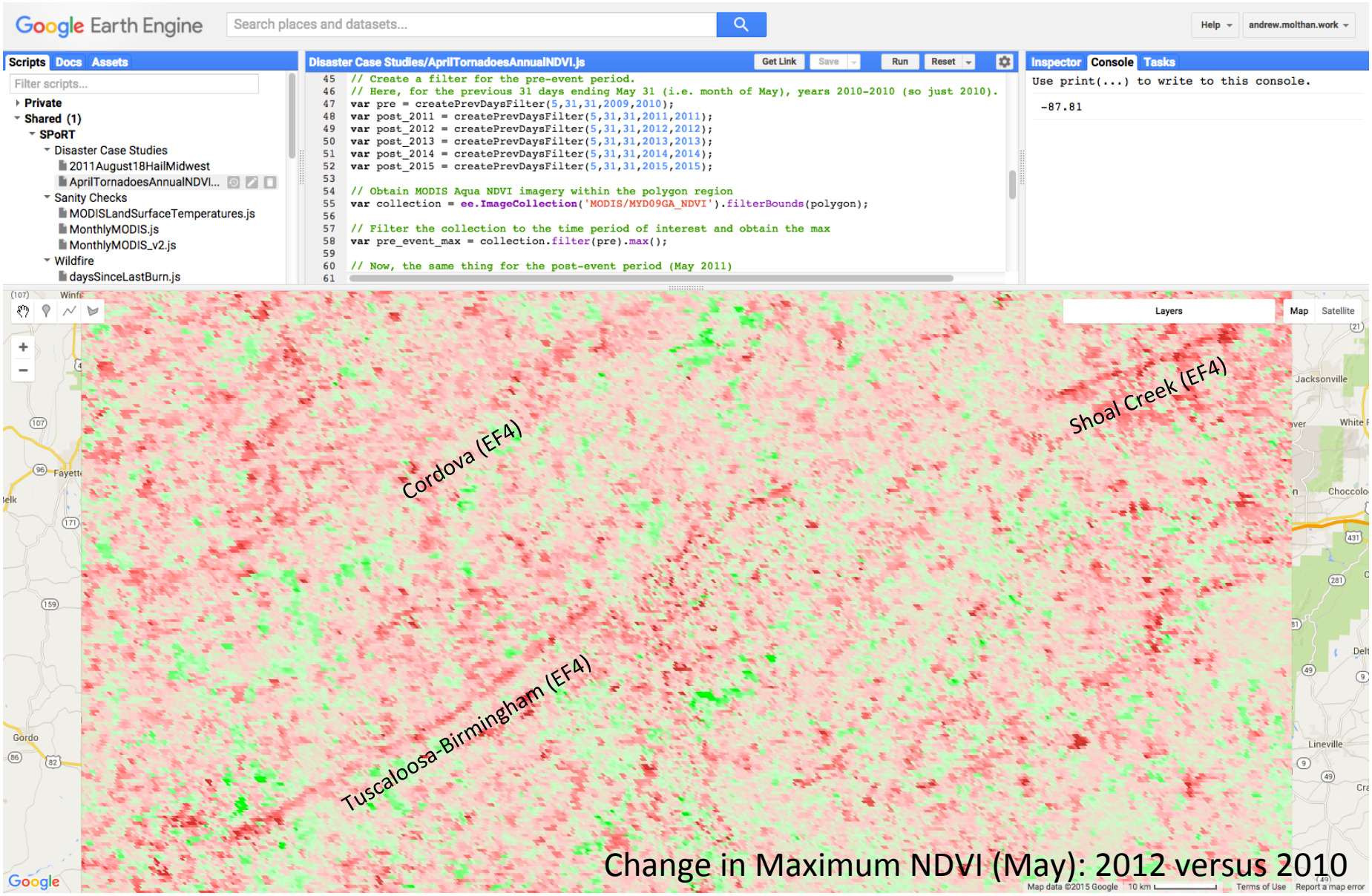
Preferable in some ways to using a web client for interaction and coding.

Applications: Land Surface Change



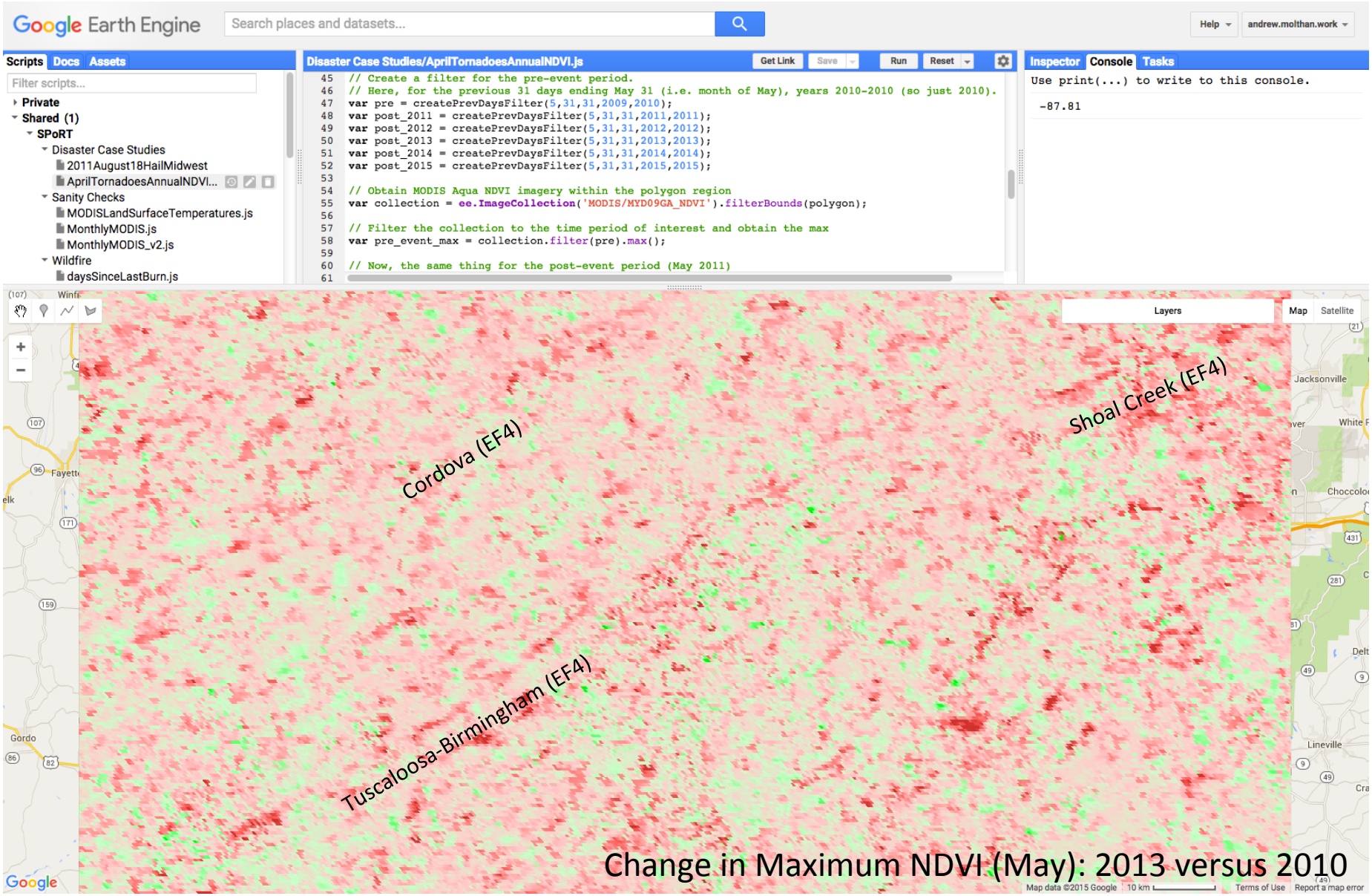
Change in Maximum NDVI (May): 2011 versus 2010

Applications: Land Surface Change



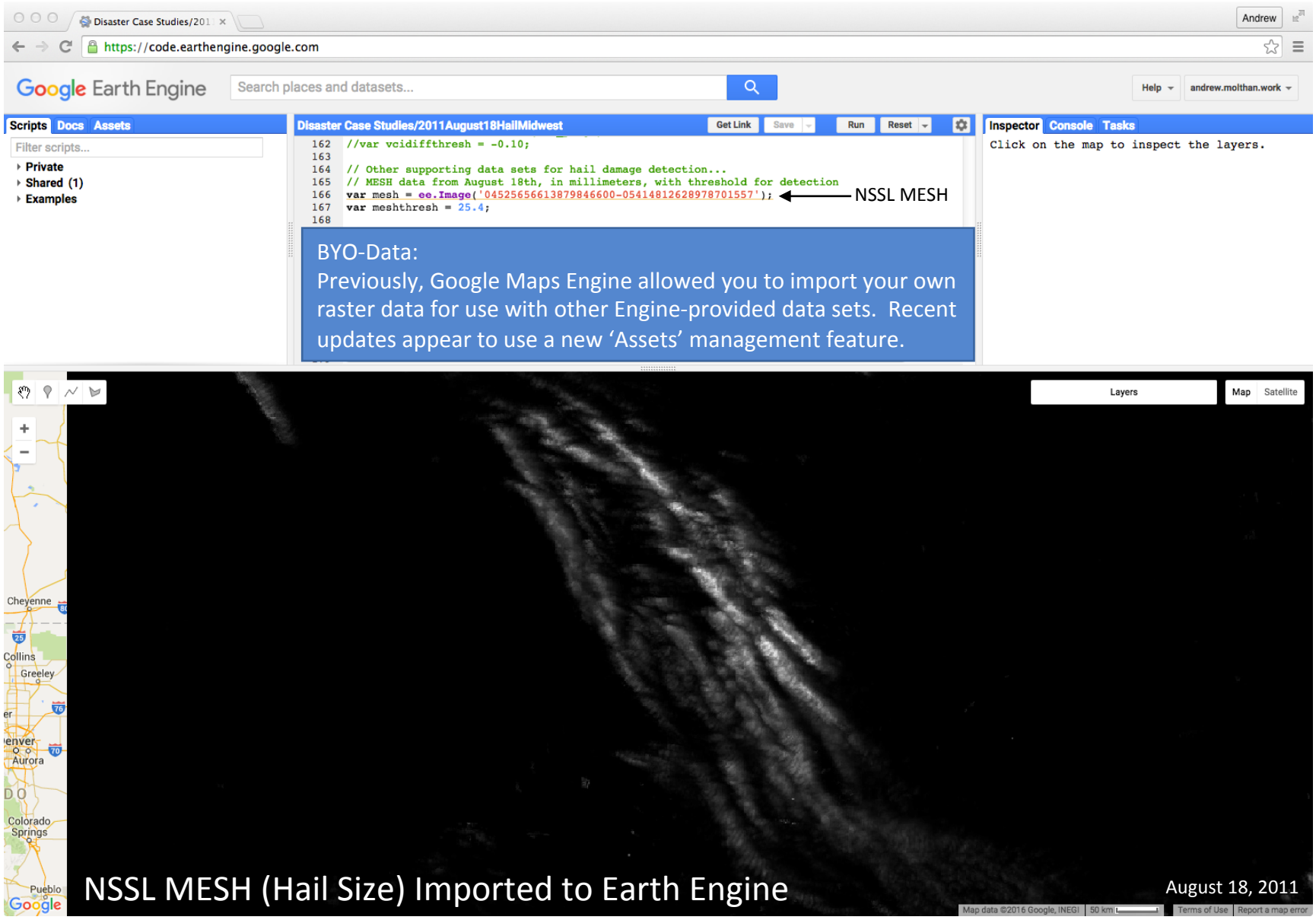
Change in Maximum NDVI (May): 2012 versus 2010

Applications: Land Surface Change

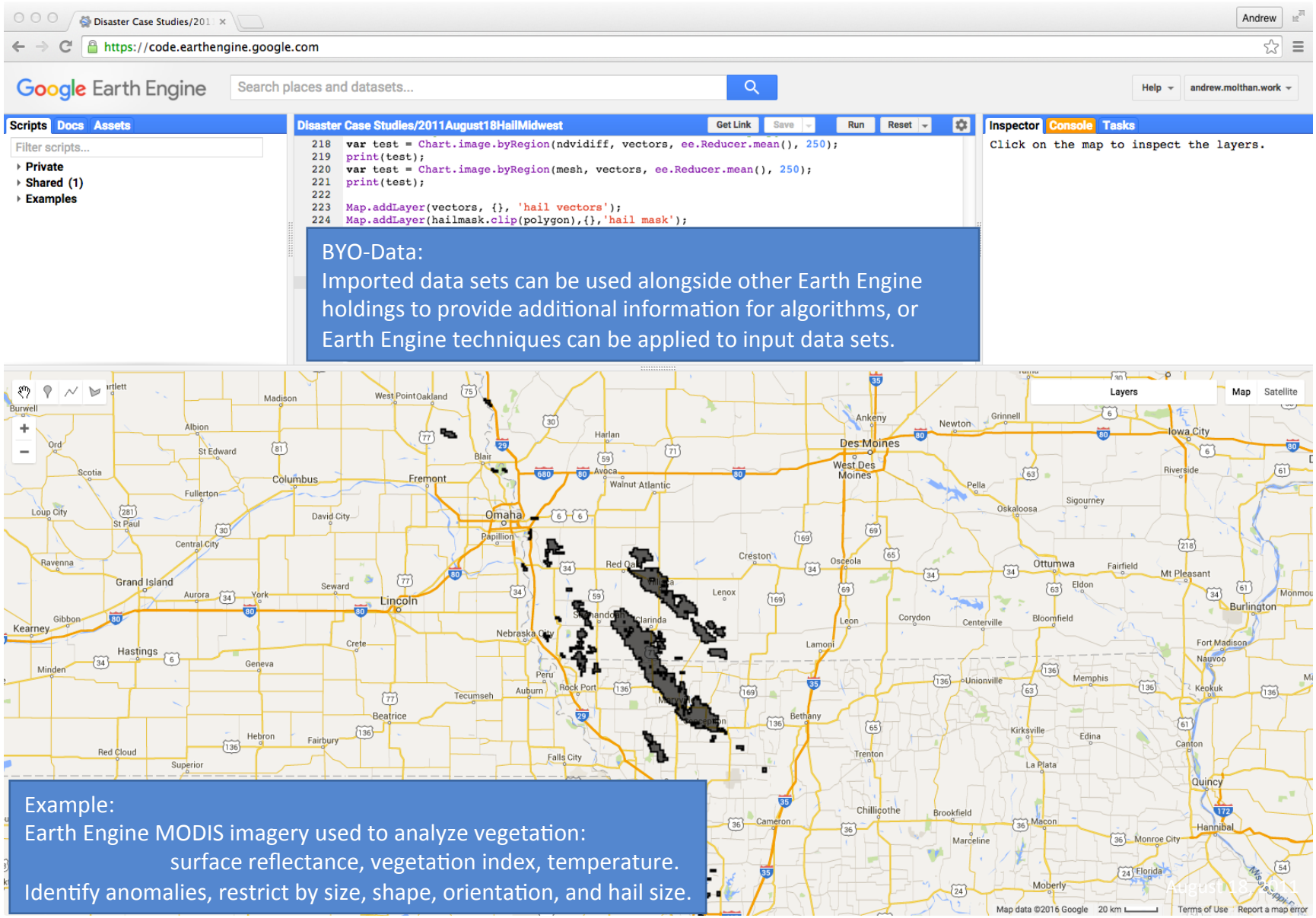


Change in Maximum NDVI (May): 2013 versus 2010

Applications: Hail Damage



Applications: Hail Damage



Based upon technique explored by Bell (2015), discussed on Thursday, 2:30, rm 252

Strengths / Weaknesses

- Strengths
  - Earth Engine data holdings are tremendous, and the streamlined Javascript/Python API is capable of processing large volumes of data very quickly.
  - Relatively simple, big data platform for exploring various remote sensing products and capabilities.
  - Numerous and increasing number of high volume data sets available that could be difficult to download and process locally.
- Weaknesses
  - Primarily focused on raster types of data – large gridded or swath products of comparable resolution.
  - Most products still require some understanding of the raw data – many appear to be a literal depositing of the source data with no additional post-processing.
    - Ex: Temperatures in scaled units, not K
  - Documentation could benefit from a few use examples, and some experiences where documentation or errors were cryptic.

Questions?

- Check out some related talks!
  - Near-Real Time Severe Weather Damage Identification Algorithm for Vegetation: Development and Early Results
    - J. Bell, University of Alabama in Huntsville
      - Thursday, 3:30 pm, 252

- andrew.molthan@nasa.gov